**1. Quick Examples of Append Multiple Items to ListS**

Following are quick examples of how to append multiple items to the [list](https://sparkbyexamples.com/python/python-list-explain/).

# Quick Examples of append items to list

# Create lists

fruits=['apple','mango','guava',]

newfruits = ['grapes','orange']

# Append items to list using +

result = fruits + newfruits

# Append list with new elements

fruits.extend(['grapes','orange'])

# Append list with tuple

fruits.extend(("pears","custard-apple"))

# Append list with set

fruits.extend({"guava","cherry","mango"})

# Append list with dictionary keys

fruits.extend({"fruit1":"pine apple","fruit2":"lemon"})

# Append list with dictionary values

fruits.extend({"fruit1":"pine apple","fruit2":"lemon"}.values())

# Append list with string

fruits.extend("papayya")

# Append list using slicing

fruits[len(fruits):] = ["guava","cherry","mango"]

Copy

**2. Append Elements to List using + Operator**

You can use the + operator in Python to append two lists into a single list. In the below example, we have two lists named fruits & newfruits and we append these lists together to get the result list. The result list contains all the elements from fruits and newfruits.

# Create lists

fruits=['apple','mango','guava',]

newfruits = ['grapes','orange']

# Append items to list

result = fruits + newfruits

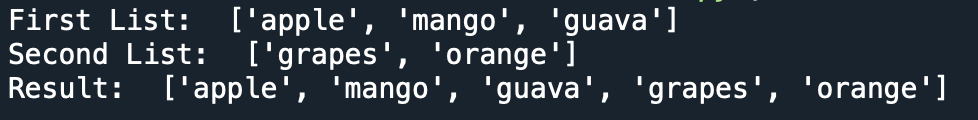
print("First List: ", fruits)

print("Second List: ", newfruits)

print("Result: ",result)

Copy

Yields below output.



**3. Append Items using append()**

By using list.append() you can append one item at a time to the python list. However, we can use this with the combination of for loop to append multiple items. Here, we iterate each element in the newfruits list and append each element to the fruits list.

# Create lists

fruits=['apple','mango','guava',]

newfruits = ['grapes','orange']

print("First List: ", fruits)

print("Second List: ", newfruits)

# Using append()

for fruit in newfruits:

fruits.append(fruit)

print("Append to First List:",fruits)

Copy

**4. Append Items to List using extend()**

The [list.extend()](https://sparkbyexamples.com/python/python-list-extend-method/) method in Python is used to append multiple items/elements to the list. This function takes the iterable as an argument and append all elements from the iterable toward the end of the existing list. Here each element is added separately to the list one after the other. If we pass a string, then it will take each character from a string as an element and appends it to the list.

**4.1 Append List to Another List**

Note that this modifies the existing list with the new elements and the elements are append at the end of the original list.

# Consider the list of strings

fruits=['apple','mango','guava','lemon']

print("Actual List: ",fruits)

# Append multiple items to the list

fruits.extend(["orange","papayya","pear"])

print("Final List: ",fruits)

# Output:

# Actual List: ['apple', 'mango', 'guava', 'lemon']

# Final List: ['apple', 'mango', 'guava', 'lemon', 'orange', 'papayya', 'pear']

Copy

Here, we had four elements in the list and appended additional 3 elements from another list. The resultant list contains a total of seven elements.

**4.2 Append Multiple Items from Tuple to List**

In this example, I will use elements from the tuple to [insert into the list](https://sparkbyexamples.com/python/python-list-insert-with-examples/). Here, my list had 2 elements to start with, and added a tuple of 3 elements, after adding a tuple, there are 5 elements in the list.

# Consider the list of strings

fruits=["papayya","pear"]

print("Actual List: ",fruits)

# Append tuple

fruits.extend(('mango','guava','lemon'))

print("Final List: ",fruits)

# Output:

# Actual List: ['papayya', 'pear']

# Final List: ['papayya', 'pear', 'mango', 'guava', 'lemon']

Copy

**4.3 Append Elements from Set**

Let’s have a list of 2 strings and append a Set of items to the list. In the below example, there are 2 elements in our list, We specified 2 elements in the set that are duplicates, As the Set doesn’t allow duplicates, it will consider 2 elements as one and append this element to the list. So the total number of elements in the list is 3.

# Consider the list of strings

fruits=['guava','lemon']

print("Actual List: ",fruits)

# Append set that has duplicates

fruits.extend({"orange","orange"})

print("Final List: ",fruits)

# Output:

# Actual List: ['guava', 'lemon']

# Final List: ['guava', 'lemon', 'orange']

Copy

**4.4 Append String**

Let’s have a list of 3 fruits and append another fruit to this list. As I mentioned in the introduction, adding a string to a list, actually appends each character from the string as an element to the list.

# Consider the list of strings

fruits=['mango','guava','lemon']

print("Actual List: ",fruits)

# Append string

fruits.extend("papayya")

print("Final List: ",fruits)

# Output:

# Actual List: ['mango', 'guava', 'lemon']

# Final List: ['mango', 'guava', 'lemon', 'p', 'a', 'p', 'a', 'y', 'y', 'a']

Copy

Previously, there are 3 elements in our list, we added one string – “papaya” to this list. As there are 4 characters in this string, the list will store each character as a single element. Now the final list holds 7 elements.

**4.5 Append Keys/Values from Dictionary**

When you use a dictionary as an argument to the list extend() method, it appends keys from the dictionary to the list. If you want to store values in to list, you need to specify values() method along with the dictionary name. It will use only values from the dictionary to the list.

Alternatively, you can also specify keys() method along with the dictionary name to insert keys. It will access only keys from the dictionary and added to the list. By default, it will store only Keys present in a dictionary.

**Example 1:** Append keys. Here, the keys in the dictionary are – “fruits1” and “fruits2” hence, these are appended to the list.

# Consider the list of strings

fruits=["orange","papayya","pear","apple"]

print("Actual List: ",fruits)

# Append dictionary of keys

fruits.extend({"fruits1":"guava","fruits2":"mango"})

print("Final List: ",fruits)

# Output:

# Actual List: ['orange', 'papayya', 'pear', 'apple']

# Final List: ['orange', 'papayya', 'pear', 'apple', 'fruits1', 'fruits2']

Copy

**Example 2:** Append Values. Here, the Values in the dictionary are – “guava” and “mango”. These are added to the list.

# Consider the list of strings

fruits=["orange","papayya","pear","apple"]

print("Actual List: ",fruits)

# Append dictionary of values

fruits.extend({"fruits1":"guava","fruits2":"mango"}.values())

print("Final List: ",fruits)

# Output:

# Actual List: ['orange', 'papayya', 'pear', 'apple']

# Final List: ['orange', 'papayya', 'pear', 'apple', 'guava', 'mango']

Copy

**5. Append Element to List using Slicing**

We can also append elements to the list through Slicing. Let’s look at the syntax that uses slicing

# Syntax

mylist1[len(mylist1):]=iterable/s

Copy

Here, mylist1 is the input list and we can append iterable to this input list. Let’s create a list of strings and append three strings to the list with the slicing approach.

# Consider the list of strings

fruits=['apple','mango',]

print("Actual List: ",fruits)

# Append list using Slicing

fruits[len(fruits):] = ["orange","papayya","pear"]

print("Final List: ",fruits)

# Output:

# Actual List: ['apple', 'mango']

# Final List: ['apple', 'mango', 'orange', 'papayya', 'pear']

Copy

**6. Conclusion**

In this article, you have learned how to append multiple items to the list in Python using the + operator, append (), extend() and slicing methods. The extend method appends elements into the list from another iterable. It takes iterable as an argument and appends all elements from it at the end of the existing list.

I am adding this NEW Page just to see if my changes will be checked against my old word file called Lists.